

GUILHERME NEVES HERKER

**DESENVOLVIMENTO DE UM ALGORITMO PARA
IMPLEMENTAÇÃO DO MÉTODO DOS ELEMENTOS
DISCRETOS**



**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
2017**

GUILHERME NEVES HERKER

**DESENVOLVIMENTO DE UM ALGORITMO PARA
IMPLEMENTAÇÃO DO MÉTODO DOS ELEMENTOS
DISCRETOS**

Monografia apresentada ao Programa de Graduação em Engenharia Aeronáutica da Universidade Federal de Uberlândia, como parte dos requisitos para a obtenção do título de **BACHAREL EM ENGENHARIA AERONÁUTICA.**

Área de concentração: Métodos Numéricos

Orientador: Prof. Dr. Francisco José de Souza

**UBERLÂNDIA – MG
2017**

GUILHERME NEVES HERKER

**DESENVOLVIMENTO DE UM ALGORITMO PARA
IMPLEMENTAÇÃO DO MÉTODO DOS ELEMENTOS
DISCRETOS**

Monografia **APROVADA** pelo Programa de
Graduação em Engenharia Aeronáutica da Universidade
Federal de Uberlândia no dia 02/08/2017.

Área de concentração: Métodos Numéricos

Banca Examinadora:

Prof. Dr. Francisco José de Souza – Orientador

Prof. Msc. Thiago Guimarães

Doutorando Pedro Ricardo

**UBERLÂNDIA – MG
2017**

RESUMO

O Método dos Elementos Discretos tem sido empregado para simular o movimento de partículas de materiais granulares e rochosos, mas tem se tornado popular como um método para representar materiais sólidos e para o estudo de problemas de fluxo, pois conduz a uma menor adoção de parâmetros de análise que os métodos em que o meio é considerado como um contínuo. Este trabalho desenvolve um algoritmo para aplicação deste método em duas dimensões, considerando força normal entre as partículas e os anteparos como sendo rígidos.

Palavras-chave: Método dos Elementos Discretos, Métodos Numéricos, Algoritmo.

LISTA DE FIGURAS

Figura 1. Algoritmo de checagem direta	12
Figura 2. Tamanho de uma célula	13
Figura 3. Células com possibilidade de haver contato	13
Figura 4. Sistema massa-mola múltiplo	14
Figura 5. Esquema do Processo iterativo	17
Figura 6 - Contato partícula-partícula (Itasca, 2004)	18
Figura 7 - Nomenclatura partícula-parede	19
Figura 8 - Eixos Coordenados	22
Figura 9. Condição de contorno - quadrado	24
Figura 10. Posição e inclinação dos anteparos	24
Figura 11. Fórmulas aplicadas nas colunas J e K no excel	26
Figura 12. Inclinação do Anteparo 1	27
Figura 13. Ângulo da partícula em relação ao Anteparo 1	28
Figura 16. Quadrado $t = 0,3s$	31
Figura 17. Quadrado $t = 0,35s$	30
Figura 18. Quadrado $t = 0,4s$	31
Figura 19. Quadrado $t = 0,5s$	30
Figura 20. Anteparos $t = 0,1s$	31
Figura 21. Anteparos $t = 0,2s$	31
Figura 22. Anteparos $t = 0,3s$	32
Figura 23. Anteparos $t = 0,4s$	32

LISTA DE TABELAS E ALGORITMOS

Tabela 1. Coeficiente de Rigidez da soja em função da umidade	15
Algoritmo 1: Algoritmo utilizando o Direct Checking	12
Algoritmo 2: Distância e comprimento da fonte de partícula.....	23
Algoritmo 3: Condições iniciais para as velocidades	23
Algoritmo 4: Condições iniciais para as partículas.....	23
Algoritmo 5: Aplicação de fórmulas para auxílio no cálculo	25
Algoritmo 6: Distância entre as partículas.....	26
Algoritmo 7: Posição do anteparo 1.....	27
Algoritmo 8: Cálculo da distância entre o anteparo 1 e a partícula.....	28

ÍNDICE

1	INTRODUÇÃO	8
1.1	OBJETIVOS	8
1.2	METODOLOGIA	9
1.3	ESCOPO E ORGANIZAÇÃO.....	9
2	REVISÃO BIBLIOGRÁFICA	10
2.1	MÉTODOS DE ITERAÇÃO.....	11
2.2	ESTIMATIVA DO PASSO DE TEMPO	14
2.2	ESTIMATIVA DO COEFICIENTE DE RIGIDEZ	14
3	FUNDAMENTOS TEÓRICOS.....	16
3.1	LEI FORÇA-DESLOCAMENTO	17
3.2	LEI DO MOVIMENTO.....	21
4	DESENVOLVIMENTO DO ALGORITMO	22
4.1	CONDIÇÕES INICIAIS.....	22
4.2	CONDIÇÕES DE CONTORNO	23
4.3	PROPRIEDADE DAS PARTÍCULAS	24
4.4	PASSO DE TEMPO	25
4.5	LÓGICA DE PROGRAMAÇÃO	25
5	RESULTADOS.....	29
6	CONCLUSÕES	33
	REFERÊNCIAS BIBLIOGRÁFICAS.....	34
	APÊNDICE A.....	35

CAPÍTULO I

INTRODUÇÃO

O Método dos Elementos Discretos (MED) tem sido empregado para simular o movimento de partículas de materiais granulares e rochosos, mas tem se tornado popular como um método para representar materiais sólidos e para o estudo de problemas de fluxo, pois conduz a uma menor adoção de parâmetros de análise que os métodos em que o meio é considerado como um contínuo.

A partir do entendimento das propriedades mecânicas microscópicas das partículas e o comportamento da interação entre elas, o MED permite avaliar de maneira macroscópica o comportamento físico e mecânico do modelo estudado.

O diferencial do método está em considerar o meio analisado como um conjunto de partículas com propriedades mecânicas particulares e geometrias definidas (meio discretizado). O mais usual é trabalhar com um conjunto de discos ou esferas, mas o método tem sido aplicado também para partículas com outras geometrias.

A associação de um programa que resolva problemas por meio do MED com softwares gráficos permite a visualização do comportamento das partículas dos materiais em estudo, sendo muito útil para a engenharia civil, engenharia mecânica, geotecnia, engenharia química, entre outras áreas do conhecimento, sendo, portanto, bem abrangente.

1.1 OBJETIVOS

O principal objetivo deste trabalho é efetuar um estudo dos conceitos básicos do Método dos Elementos Discretos (MED) e de sua implementação computacional, particularizando para elementos circulares em duas dimensões. O trabalho culmina com a criação de um programa que realiza os cálculos das variáveis dinâmicas dos elementos e proporciona uma visualização dos mesmos.

1.2 METODOLOGIA

Para a simulação do comportamento das partículas, será desenvolvido um algoritmo em Excel VBA, baseado na teoria do MED. Várias condições de contorno diferentes serão impostas para analisar a influência de cada condição no resultado do movimento das partículas. Um pós-processamento dos dados será feito com o mesmo software, para que seja possível visualizar o que ocorre com as partículas.

1.3 ESCOPO E ORGANIZAÇÃO

Este trabalho será composto de 6 capítulos e um apêndice.

No presente capítulo são apresentados a relevância da pesquisa, objetivos e um resumo da metodologia empregada na mesma.

No **Capítulo 2** será apresentada uma breve revisão bibliográfica sobre as aplicações e tipos de algoritmos já empregados para resolução de problemas utilizando o MED.

O **Capítulo 3** apresenta detalhes sobre o método dos elementos discretos com a lógica por trás dos elementos básicos do modelo (partículas e paredes), modelos de contato, modelos de amortecimento e outras informações relevantes para a simulação do comportamento das partículas.

No **Capítulo 4** será discutido com mais detalhes como o método foi aplicado.

No **Capítulo 5** serão apresentados os resultados das simulações realizadas com as respectivas análises.

No **Capítulo 6** serão apresentadas conclusões e sugestões para pesquisas futuras.

No **Apêndice A**, encontra-se o código desenvolvido em VBA.

CAPÍTULO II

REVISÃO BIBLIOGRÁFICA

O Método dos Elementos Discretos (MED) encontra-se em um estágio de desenvolvimento um pouco anterior aos outros métodos de soluções aproximadas, como o Método dos Elementos Finitos (MEF) e o Método dos Elementos de Contorno (MEC), por exemplo.

As características básicas do MED foram inicialmente propostas por Cundall e Strack (1979) e vem se desenvolvendo com novas formulações apresentadas e publicadas em periódicos e congressos da área.

Para uso eficiente de uma ferramenta numérica como o MED é preciso, no entanto, identificar para quais tipos de problemas o seu uso é mais recomendado. O método é mais adequado para lidar com problemas de natureza descontínua e ao mesmo tempo com grandes níveis de deformação. Devido à essas características, ele tem sido utilizado em problemas de diversas áreas como em aplicações industriais em grande escala (Cleary, 2004).

Mais especificamente na engenharia geotécnica, suas funcionalidades têm sido utilizadas em problemas de mecânica das rochas (Cundall; Tannant e Wang, 2004). No campo da mecânica dos solos, o seu uso é bastante variado, como por exemplo, na localização de gradientes de deformação em meios granulares (Kuhn, 2005), comportamento de argilas (Yao; Anandarajah, 2003), interação fluido-partículas (Preece et al., 1999; Cook e al., 2004), etc.

No MED, a disposição das partículas dentro do conjunto global do sistema é aleatória, sendo possível formar meios com diferentes tamanhos de partículas distribuídas ao longo do conjunto, idealizando deste jeito, a natureza granular dos meios que usualmente é estudada através desta técnica numérica. Principalmente podem-se distinguir as seguintes propriedades básicas que definem de forma global deste método de análise numérica:

- As partículas como elementos discretos que no seu conjunto conformam o sistema complexo de partículas.
- Estes elementos distintos deslocam-se independentemente um do outro e interagem entre eles na zona de contato.

- Neste método, para cada partícula, usa-se a mecânica do meio rígido e os elementos discretos consideram-se elementos rígidos entre eles.

O comportamento global do material é governado pelo modelo constitutivo aplicado nos contatos entre as partículas. O cálculo executado no método integra a aplicação da segunda Lei de Newton para as partículas e a Lei de Força – Deslocamento para os contatos.

1. Segunda Lei de Newton: Determina o movimento de cada partícula a partir das forças de contato e as forças aplicadas nas partículas.

2. Lei de Força – Deslocamento: Atualiza as forças de contato surgidas do movimento relativo de cada contato.

A caracterização dos contatos neste modelo se descreve pelos seguintes elementos mecânicos:

- Molas: Os elementos molas descrevem a fase do comportamento elástico do meio na zona de contato entre cada partícula. Este comportamento elástico é caracterizado por duas molas, uma na direção normal ao contato e outra na direção tangencial, as quais correspondem com a decomposição das forças de contato que são usadas na formulação do método.
- Amortecedores: São elementos que levam em conta a viscosidade do meio que é simulado. Na formulação estabelecida podem usar-se vários modelos de contato que podem ser delimitados em modelos de contato viscoso e não viscoso o que permite aplicar o modelo para um maior número de problemas mecânicos, tanto elásticos como viscoelásticos.
- Elementos de Atrito: Estes elementos descrevem a rotura da coesão e a falha do material na zona de contato entre cada partícula. Quando no contato se produz a ruptura, a partícula desprende-se do meio.

2.1 MÉTODOS DE ITERAÇÃO

2.1.1 DIRECT CHECKING OU CHECAGEM DIRETA

Esse método de detecção consiste em nada mais do que calcular todas as possíveis colisões entre os elementos (Figura 1).

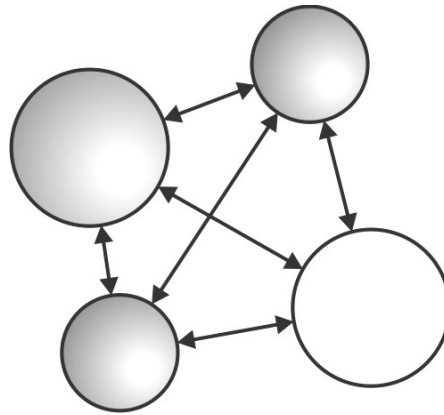


Figura 1. Algoritmo de checagem direta (THIMOUNIER, 2009)

Assim o número de cálculos feitos será assintoticamente proporcional a n^2 , o que pode tornar esse cálculo muito pesado para um grande número de elementos. Este tipo de abordagem tem a vantagem de sua implementação ser extremamente simples. Porém, tem utilidade apenas para problemas com quantidade pequena de elementos.

Algoritmo 1: Algoritmo utilizando o Direct Checking

```

For i = 1 To n
  For j = i + 1 To n
    'Calculo da colisão entre os elementos i e j
  Next j
Next i

```

2.1.2 DIRECT MAPPING OU MAPEAMENTO DIRETO

Este método baseia-se no mapeamento espacial do domínio, dividindo-o em células cúbicas de tamanhos iguais. O lado das células deverá ser escolhido de forma que qualquer elemento possa estar contido em uma célula (MUNJIZA, 2004).

Assim o lado de uma célula cúbica é igual ao diâmetro do círculo que circunscreve o maior elemento discreto, conforme a figura a baixo:

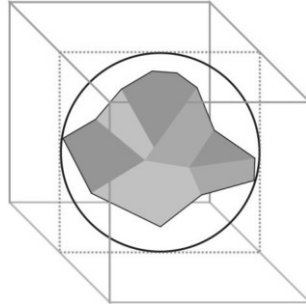


Figura 2. Tamanho de uma célula (THIMOUNIER, 2009)

O algoritmo Munjiza é dividido em duas etapas principais: o mapeamento de cada elemento em sua célula correspondente e a procura dos possíveis elementos que podem estar em contato entre si. Para encontrar os possíveis contatos com um determinado elemento checam-se somente as células vizinhas à célula desse elemento e a sua própria célula (Figura 4).

Para essa consideração ser válida, definimos o tamanho da célula como aquele que pode conter o maior elemento discreto. Após a constatação da possibilidade de contato entre partículas pertencentes às células vizinhas ou a mesma célula, realiza-se outro teste de detecção entre as partículas para que se confirme o contato, devendo esse segundo teste depender da geometria dos elementos.

Munjiza mostra que esse algoritmo de mapeamento direto para busca de contatos entre partículas é assintoticamente proporcional ao número de elementos, o que torna esse algoritmo muito mais apropriado para problemas com um grande número de elementos.

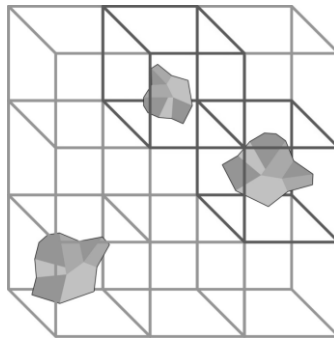


Figura 3. Células com possibilidade de haver contato (THIMOUNIER, 2009)

Esse algoritmo apresenta somente duas limitações, a primeira é que para partículas de tamanhos bem variáveis ele deixa de ser assintoticamente proporcional ao número de elementos, e a segunda é que pode apresentar uso excessivo de memória computacional caso o tipo de problema estudado possua elementos discretos muitos dispersos no domínio estudado.

2.2 ESTIMATIVA DO PASSO DE TEMPO

Para estimar um intervalo de tempo suficientemente pequeno, e garantir que sistema seja estável, e portanto, corresponda com a realidade, é necessário considerar o menor período de vibração natural das partículas.

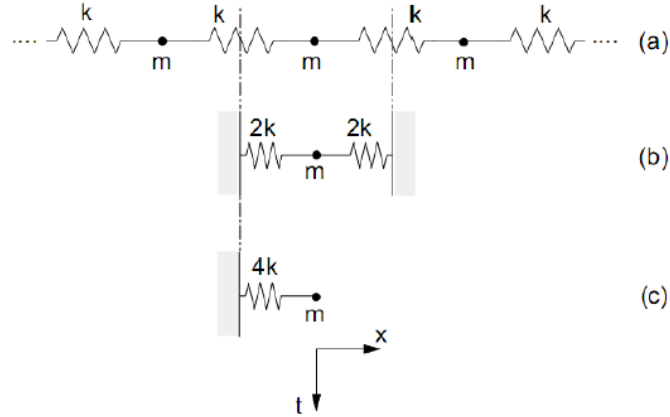


Figura 4. Sistema massa-mola múltiplo (ITASCA, 2004)

Para um número infinito de pontos massa-mola (Figura 5a), o menor período de vibração ocorrerá quando as massas estiverem movimentando-se em sentido oposto e de forma sincronizada. Este movimento faz com que o centro das molas seja estacionário tal que o movimento de um único ponto de massa possa ser descrito como na Figura 3.5 (b) e (c). O intervalo de tempo crítico será então:

$$t_{crit} = 2\sqrt{\frac{m}{4k}} = \sqrt{\frac{m}{k}} \quad (2.1)$$

onde, k pode ser considerado como o maior valor entre K^n e K^s .

2.2 ESTIMATIVA DO COEFICIENTE DE RIGIDEZ

Utilizando a tabela a baixo como estimativa para o coeficiente de rigidez da soja [5], pode-se estimar como sendo a média em cada teor de água, portanto $K_{soja} \cong 140$ [N/mm].

Tabela 1. Coeficiente de Rigidez da soja em função da umidade

Teor de Água (b.s.)	Tangente Máxima (N mm ⁻¹)	Secante Máxima (N mm ⁻¹)
0,315	45,321	37,586
0,252	58,885	43,452
0,217	67,769	50,720
0,168	124,720	96,442
0,135	172,971	132,639
0,093	367,778	252,080

CAPÍTULO III

FUNDAMENTOS TEÓRICOS

A formulação estabelecida para o desenvolvimento do modelo pelo Método dos Elementos Discretos adota uma série de hipóteses que permitem simplificar o problema real descartando os aspectos menos significativos e permitindo estabelecer um modelo físico e matemático do problema em estudo. No caso deste trabalho, serão consideradas apenas a força normal entre as partículas. As hipóteses que são estabelecidas são as seguintes:

1. As partículas são consideradas corpos rígidos;
2. O contato acontece num ponto ou numa área muito pequena do contato entre cada partícula;
3. No contato existe uma tolerância de penetração entre as partículas. A magnitude desta é relacionada com a força de contato através da Lei Força-Deslocamento e é muito pequena em relação ao tamanho das partículas;
4. Todas as partículas são circulares. Em 2D são utilizados cilindros e em 3D esferas;
5. A geração de um meio empregando elementos discretos deve ser aleatória e os diâmetros dos mesmos devem ser tratados de forma similar (posição e diâmetro dos elementos distintos são aleatórios);
6. Nas uniões entre partículas considera-se que existem contatos entre cada partícula.

Na formulação podem-se incluir elementos rígidos ou paredes, as quais podem ter velocidade, deslocamentos ou forças. Estas paredes dão as condições de contorno às partículas, e podem compactar um arranjo de partículas. No método dos elementos discretos, identifica-se um ciclo de cálculo básico, apresentado na Figura 5, onde são executadas as principais etapas do método. Para o desenvolvimento de sua formulação e de sua implementação é interessante abordar e desenvolver cada etapa de cálculo individualmente, o que será feito a seguir:

No MED a interação entre partículas é tratada como um processo dinâmico com estados de equilíbrio sendo atingidos sempre que houver equilíbrio de forças. As forças de contato e deslocamentos de um conjunto de partículas sujeito a um determinado estado de tensões é encontrado por meio do

monitoramento individual do movimento das partículas. Os movimentos resultam da propagação de perturbações através do conjunto de partículas causadas por movimentos e forças aplicados em paredes e/ou partículas.

O comportamento dinâmico é representado numericamente por um algoritmo de solução explícita no tempo, utilizando um esquema de diferenças finitas centrais para integrar acelerações e velocidades. O MED é baseado na ideia de que o passo de tempo escolhido é tão pequeno que, durante um único intervalo de tempo, as perturbações podem propagar-se somente para a partícula vizinha. Então, em todos os tempos, as forças atuando em qualquer partícula podem ser determinadas exclusivamente pelas interações com as partículas com as quais ela estiver em contato. O uso de um esquema numérico explícito torna possível simular interações não lineares de um grande número de partículas sem exigir grande quantidade de memória do computador por restringir o processo iterativo às partículas vizinhas.

O ciclo de cálculo do MED alterna entre a aplicação da segunda Lei de Newton às partículas e a lei de força-deslocamento nos contatos. A segunda Lei de Newton é usada na determinação do movimento de cada partícula, resultante do contato e forças de corpo atuando nesta, enquanto que a lei de força-deslocamento é usada para atualizar as forças de contato resultantes do movimento relativo em cada contato. No caso deste algoritmo as paredes são consideradas rígidas.

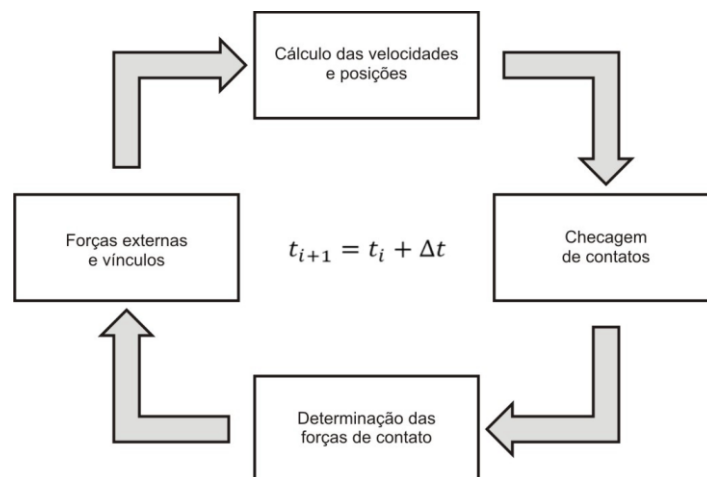


Figura 5. Esquema do Processo iterativo

3.1 LEI FORÇA-DESLOCAMENTO

A Lei Força-Deslocamento é aplicada no contato de duas entidades. O contato pode ser formado por duas partículas ou por uma partícula e uma parede sendo a sua posição representada como $x_i^{[C]}$. O

contato encontra-se em um plano com vetor normal, sendo este definido pela reta que une os centros das duas partículas (da partícula A para B). O vetor normal é:

$$\mathbf{n}_i = \frac{\mathbf{x}_i^{[B]} - \mathbf{x}_i^{[A]}}{d} \quad (3.1)$$

Sendo d a distância entre os centros das duas partículas em contato. Este valor pode ser determinado por:

$$d = |\mathbf{x}_i^{[B]} - \mathbf{x}_i^{[A]}| = \sqrt{(\mathbf{x}_i^{[B]} - \mathbf{x}_i^{[A]}) \cdot (\mathbf{x}_i^{[B]} - \mathbf{x}_i^{[A]})} \quad (3.2)$$

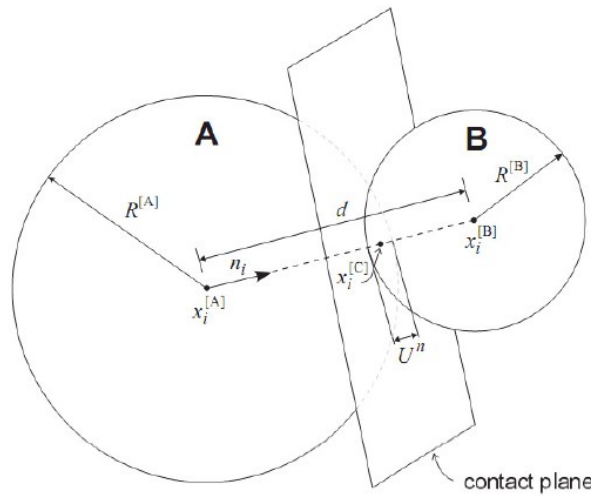


Figura 6 - Contato partícula-partícula (Itasca, 2004)

Se o contato for partícula-parede, a normal terá a direção da reta de menor distância entre o centro da partícula e a parede. A superposição de uma partícula sobre outra é definida pelo deslocamento relativo das partículas na direção normal e é dada pela Equação 3.3. A superposição é representada por U^n . Estas definições podem ser vistas nas Figuras 7 e 8.

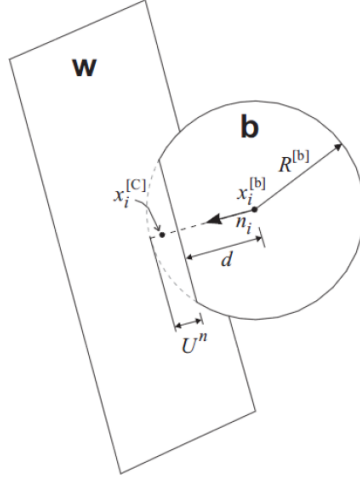


Figura 7 - Nomenclatura partícula-parede

A superposição das partículas em contato pode-se determinar, conhecendo os raios das entidades em contato e a distância inicial entre elas.

$$U^n = \begin{cases} R^{[A]} + R^{[B]} - d, & (\text{partícula - partícula}) \\ R^{[b]} - d, & (\text{partícula - parede}) \end{cases} \quad (3.3)$$

Sendo $R^{[\alpha]}$ o raio da partícula α . Após a definição destas variáveis, a posição do contato é determinada por:

$$\mathbf{x}_i^{[c]} = \begin{cases} \mathbf{x}_i^{[A]} + \left(R^{[A]} - \frac{1}{2} U^n \right) \mathbf{n}_i, & (\text{partícula - partícula}) \\ \mathbf{x}_i^{[b]} + \left(R^{[b]} - \frac{1}{2} U^n \right) \mathbf{n}_i, & (\text{partícula - parede}) \end{cases} \quad (3.4)$$

A força gerada no contato é baseada na superposição de partículas. Esta pode ser no sentido normal ou cisalhante. A força total no contato é a soma destas, sendo a sua magnitude dependente da lei de rigidez adotada.

$$\mathbf{F}_i = \mathbf{F}_i^n + \mathbf{F}_i^s \quad (3.5)$$

As partículas atuam como se fossem unidas por molas nos seus contatos. Baseado neste critério, a força gerada no contato será o produto da deformação da mola pela sua rigidez. Assim, o deslocamento no sentido normal que seria a deformação da mola nessa orientação é a superposição. Sendo a rigidez normal do contato igual a K_n , a força normal é:

$$\mathbf{F}_i^n = K^n U^n \mathbf{n}_i \quad (3.6)$$

A rigidez cisalhante, K^s , por outro lado, é um módulo tangente e está relacionada à força e deslocamento incrementais.

A força de contato cisalhante é calculada de modo incremental. Quando um contato é formado, a força cisalhante total é nula. Cada incremento deslocamento cisalhante relativo subsequente, resulta em incremento de força cisalhante (elástica) adicionado ao valor atual. O movimento do contato é considerado durante o processo por meio da atualização de $\mathbf{n}_i$ e $\mathbf{x}_i^{[C]}$ a cada passo de tempo.

O movimento cisalhante relativo no contato, ou a velocidade de cisalhamento no contato, V_s (definida como a velocidade de cisalhamento da partícula B relativa à partícula A no contato partícula-partícula, e como a velocidade de cisalhamento da partícula no ponto de contato com a parede para o contato partícula-parede), é dada por:

$$V_s = \left(\dot{\mathbf{x}}_k^{[\Phi^2]} - \dot{\mathbf{x}}_k^{[\Phi^1]} \right) \cdot \mathbf{n}_k \quad (3.7)$$

onde $\dot{\mathbf{x}}_k^{[\Phi^j]}$ e $\omega_3^{[\Phi^j]}$ são as velocidades translacional e rotacional, respectivamente, da entidade Φ^j dada por:

$$\{\Phi^1, \Phi^2\} = \begin{cases} \{A, B\}, & (\text{partícula - partícula}) \\ \{b, w\}, & (\text{partícula - parede}) \end{cases} \quad (3.8)$$

e $\mathbf{t}_i = \{-\mathbf{n}_2, \mathbf{n}_1\}$.

O componente de cisalhamento do incremento de deslocamento cisalhante que ocorre num passo de tempo Δt é calculado como

$$\Delta U^s = V_s \Delta t \quad (3.9)$$

e é utilizado para calcular o incremento de força elástica cisalhante:

$$\Delta F^s = -K^s \Delta U^s \quad (3.10)$$

onde K^s é a rigidez ao cisalhamento no contato. O seu valor é determinado pelo modelo de rigidez de contato utilizado.

A nova força de contato cisalhante é encontrada a partir da soma da força cisalhante existente no início do passo de tempo com o incremento de força cisalhante elástica

$$F^s \leftarrow F^s + \Delta F^s \leq \mu F^n \quad (3.11)$$

onde μ é o coeficiente de atrito.

3.2 LEI DO MOVIMENTO

Após ter formulado as equações para cálculo das forças, será empregada a Segunda Lei de Newton para calcular a aceleração de cada partícula, e suas consequentes velocidade e posição. A equação vetorial a seguir expressa essa relação:

$$\mathbf{F}_i = m_i \frac{d^2 \mathbf{s}_i}{dt^2} \quad (3.12)$$

onde $\mathbf{F}_i$ representa a resultante das forças atuantes sobre a partícula i , m_i a massa da partícula i e $\mathbf{s}_i$ o respectivo vetor posição.

Existem diversos métodos de solução, os quais podem ser analíticos ou numéricos, sendo estes mais versáteis e, portanto, o que será utilizado neste trabalho. Assim, se quisermos saber as posições e velocidades de uma partícula entre um instante inicial t_0 e final t_f , poderemos dividir o intervalo $[t_0; t_f]$ em N partes iguais a Δt e calcular as velocidades e posições nos instantes usando as acelerações no instante anterior.

Para tanto, utilizaremos a expansão em série de Taylor, para determinar a posição e velocidade da partícula $i+1$, num instante $t_i = n\Delta t + t_0$:

$$s_{i+1} = s_i + v_i \Delta t + a_i \frac{(\Delta t)^2}{2} + a_i' \frac{(\Delta t)^3}{6} + \dots \quad (3.13)$$

$$v_{i+1} = v_i + a_i \Delta t + a_i' \frac{(\Delta t)^2}{2} + \dots \quad (3.14)$$

Nas equações 3.13 e 3.14, desprezaremos os termos de ordem 3 ou superiores e de ordem 2 ou superiores, respectivamente. O erro cometido a cada iteração passa a ser proporcional à Δt^3 , no primeiro caso, e Δt^2 no segundo caso, desde que Δt seja suficientemente pequeno. E, como temos que calcular N valores de s e v , o erro total cometido pode ser estimado como sendo proporcional a:

$$\begin{aligned} E_s &\propto N(\Delta t^3) \propto \Delta t^2 \\ E_v &\propto N(\Delta t^2) \propto \Delta t \end{aligned} \quad (3.15)$$

CAPÍTULO IV

DESENVOLVIMENTO DO ALGORITMO

Para simular a colisão entre as partículas, é necessário, inicialmente, estabelecer as condições iniciais, ou seja, posição, velocidade e aceleração de cada partícula, bem como, as condições de contorno, ou seja, os anteparos. É necessário também estabelecer o tamanho das partículas e sua geometria e também as propriedades físicas, ou seja, densidade, módulo de elasticidade, coeficiente de atrito.

4.1 CONDIÇÕES INICIAIS

Para facilitar a análise, serão fixados eixos coordenados - que serão usados como referência para este trabalho como um todo, de acordo com a figura a seguir:



Figura 8 - Eixos Coordenados

Definido o plano cartesiano, pode-se fixar a posição, velocidade e aceleração das partículas, explicitadas a seguir:

- a. **Posição:** a posição de cada partícula em y , será inicialmente igual a zero. Já a posição em x , será definida pela partícula anterior, um após a outra, de forma que não ultrapasse o tamanho máximo, definido como *SrcWidth*, e a distância mínima horizontal, *sl*.

Algoritmo 2: Distância e comprimento da fonte de partícula

```
s1 = 1                      'Source left
SrcWidth = 1                'Source Width
```

- b. **Velocidade:** a velocidade de cada partícula contará com um componente aleatório na componente vertical em um valor fixo para componente horizontal. A forma de calcular a componente aleatória será: um número entre zero e um, multiplicado por 0,1, fixado com 6 casas decimais.

Algoritmo 3: Condições iniciais para as velocidades

```
.Range("vi").Cells(i, 1) = cte          'velocidade em x
.Range("vj").Cells(i, 1) = Round(Rnd() * 0.1, 6)  'velocidade em y
```

- c. **Aceleração:** a aceleração em x será zero e em y , será a aceleração devido à atração gravitacional, igual à $g = 9,80665 \text{ m/s}^2$.

Realizando esta operação para todas as partículas tem-se que:

Algoritmo 4: Condições iniciais para as partículas

```
For i = 1 To n
    .Range("n_part").Cells(i, 1) = i
    .Range("t").Cells(i, 1) = 0
    .Range("diam").Cells(i, 1) = Round(d_min + Rnd() * (d_max - d_min), 6)
    .Range("sj").Cells(i, 1) = 0
    .Range("vi").Cells(i, 1) = 1
    .Range("vj").Cells(i, 1) = Round(Rnd() * 0.1, 6)
    .Range("ai").Cells(i, 1) = 0
    .Range("aj").Cells(i, 1) = g
Next i
```

4.2 CONDIÇÕES DE CONTORNO

- i. QUADRADO

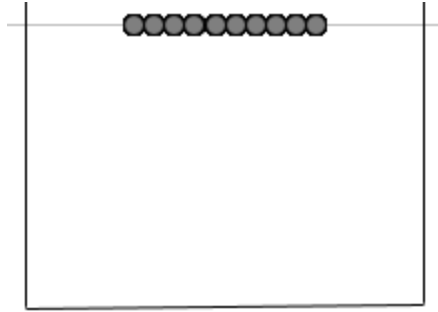


Figura 9. Condição de contorno - quadrado

ii. ANTEPAROS INCLINADOS

As condições de contorno com anteparos, são mostradas a seguir:

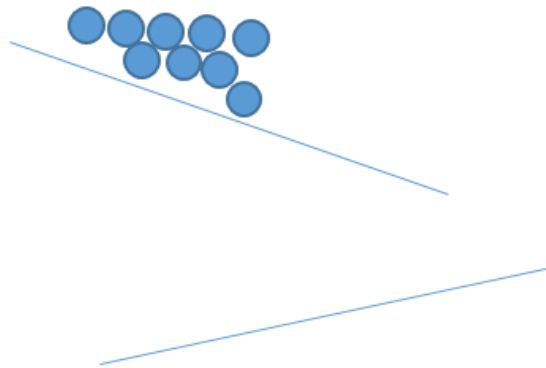


Figura 10. Posição e inclinação dos anteparos

4.3 PROPRIEDADE DAS PARTÍCULAS

Diâmetro médio: $D = 0,005 \text{ m}$

Densidade: $\rho = 1,1 \times 10^3 \text{ Kg/m}^3$

Coeficiente de Rigidez: $K^n = 140 \times 10^3 \text{ N/m}$

Coeficiente de Atrito: $\mu = 0,1$.

4.4 PASSO DE TEMPO

Como mostrado no Capítulo 2, o passo de tempo crítico, pode ser calculado pela Equação 2.1, e pela referência [2], será usado uma fração de tempo crítico, dado pela equação a baixo:

$$\Delta t = 0,05 \times t_{crit} \quad (4.1)$$

Portanto, o passo de tempo considerado será:

$$\Delta t = 0,005 \times \sqrt{\frac{m}{K^n}} = 0,005 \times \sqrt{\frac{\left(\frac{\pi}{6} D^3\right) \rho}{K^n}} = 3,2 \times 10^{-6} s \quad (4.2)$$

4.5 LÓGICA DE PROGRAMAÇÃO

Como mostrado na Seção 2.1 do Capítulo 2, existem duas técnicas para se realizar a distância entre as partículas, o *direct checking* e o *direct mapping*. Neste trabalho será utilizado o primeiro caso, porém ao invés de realizar um loop entre todas as partículas, será utilizado uma função aplicada na célula correspondente às partículas.

Algoritmo 5: Aplicação de fórmulas para auxílio no cálculo

```
ws2D.Range("J2").Formula = "=ROUND(SQRT((D2-$D$" & _  
    linha_part + 1 & _  
    ")^2+(E2-$E$" & linha_part + 1 & _  
    ")^2), 3) "  
ws2D.Range("K2").Formula = "=IF(J2=0,FALSE,J2<((C2+$C$" & _  
    linha_part + 1 & ")/2) "  
ws2D.Range("J2:K2").AutoFill Destination:=ws2D.Range("J2:K" & _  
    qnt_part + 1)
```

Na ‘coluna J’ é feito o cálculo da distância entre os centros das partículas e na ‘coluna K’ se esta distância é menor que a soma dos raios.

J	K
distP [m]	distC [m]
0,9000	FALSE
0,8000	FALSE
0,7000	FALSE
0,6000	FALSE
0,5000	FALSE
0,4000	FALSE
0,3000	FALSE
0,2000	FALSE
0,1000	FALSE
0,0000	FALSE

Figura 11. Fórmulas aplicadas nas colunas J e K no excel

O objetivo ao realizar este procedimento é utilizar uma outra função que busca somente os valores TRUE, nas células, o que significa que as partículas estão sobrepostas.

Algoritmo 6: Distância entre as partículas

```

' ----- partícula-partícula -----
cont_contato = Application.WorksheetFunction.CountIf(ws2D.Columns("K"), True)
If cont_contato > 0 Then
    linha = 1
    For c = 1 To cont_contato

        linha = ws2D.Columns("K").Find(What:=True, LookIn:=xlValues, _
            after:=ws2D.Range("K" & linha)).Row

        Fn = Kn * ws2D.Range("J" & linha) / 2

        x = ws2D.Range("si").Cells(linha_part, 1) - ws2D.Range("si").Cells(linha - 1, 1)
        y = ws2D.Range("sj").Cells(linha_part, 1) - ws2D.Range("sj").Cells(linha - 1, 1)
        dist_centro = ws2D.Range("J" & linha)

        Fni = Fn * x / dist_centro
        Fnj = Fn * y / dist_centro

    Next c
End If

```

As forças aplicadas em cada partícula, dependerá das posições absolutas de cada uma, do tamanho dos diâmetros e da consequente relação entre os centros. Quanto mais próximo os centros estiverem, maior serão as forças F_{ni} e F_{nj} . Como pode ser visto na figura a cima, é feito um loop na quantidade de partículas em contato, e as forças vertical e horizontal serão somadas.

A direção da força, indicada pelo sinal resultante, positivo ou negativo, dependerá da posição entre as partículas, calculado pelas variáveis x e y .

Com relação ao contato entre as partículas e os anteparos, a ideia é a mesma da realizada anteriormente, porém será assumido que os anteparos não se deformam, e que há atrito entre eles e as partículas.

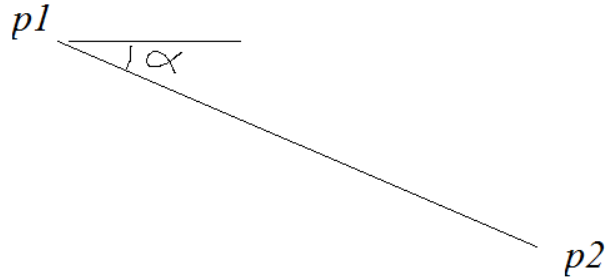


Figura 12. Inclinação do Anteparo 1

Para exemplificar, considere as variáveis $a1_p1x$ e $a1_p1y$ como sendo a coordenada superior do anteparo 1 e $a1_p2x$ e $a1_p2y$ a outra coordenada. A inclinação do anteparo, em relação à horizontal é calculada pela variável $a1_alpha$.

$$\alpha = \text{tg}^{-1} \left(\frac{p_{2,y} - p_{1,y}}{p_{2,x} - p_{1,x}} \right) \quad (4.3)$$

Algoritmo 7: Posição do anteparo 1

```
' ----- Anteparo 1 -----
a1_p1x = 0
a1_p1y = 0
a1_p2x = 4
a1_p2y = 2
a1_alpha = Atn((a1_p2y - a1_p1y) / (a1_p2x - a1_p1x))
'
```

Para calcular a distância entre o centro da partícula e o anteparo, será considerado o ângulo da reta que une o centro da partícula com o ponto p1 e a horizontal, dado pela figura a seguir:

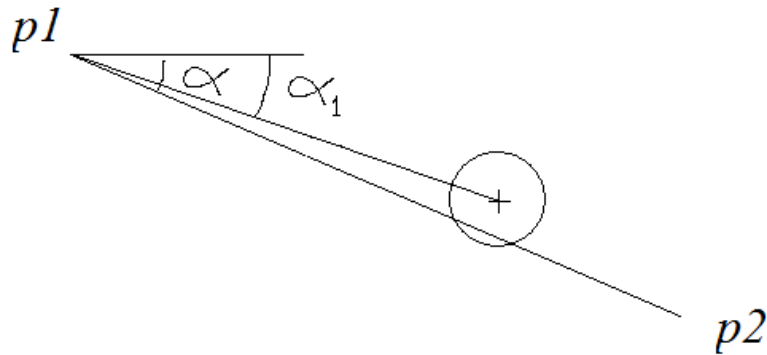


Figura 13. Ângulo da partícula em relação ao Anteparo 1

A distância entre o centro da partícula e o anteparo é calculada como sendo a projeção da distância entre o centro da partícula e o ponto $p1$, pelo ângulo $\alpha - \alpha_1$. Se essa distância for menor que o raio da partícula em análise, haverá contato e a força resultante é dada pela Equação 3.6.

Algoritmo 8: Cálculo da distância entre o anteparo 1 e a partícula

```
' ----- partícula-parede -----
' Anteparo 1
al_alpha1 = Atn((sj0 - al_p1y) / si0)
dist = (((si0 - al_p1x) ^ 2 + (sj0 - al_p2y) ^ 2) ^ 0.5) * Sin(al_alpha - al_alpha1)
If dist < (ws2D.Range("diam").Cells(linha_part, 1) / 2) And sj0 >= al_p1y Then

    Fn = Kn * dist
    Fat = u * Fn

    Fni = Fni + Fn * Sin(al_alpha)
    Fnj = Fnj - Fn * Cos(al_alpha)

End If
```

É realizada a mesma operação para os demais anteparos, e ao final do processo, é feita uma soma das forças, para que a Equação 3.12 possa ser aplicada.

CAPÍTULO V

RESULTADOS

Para analisar a influência das condições de contorno, serão simuladas algumas geometrias com condições iniciais específicas para cada caso.

i. QUADRADO

O caso mais simples de simulação, é o caso das partículas caindo com aceleração da gravidade. Neste caso, foram simuladas partículas com diâmetro e propriedades da soja. As velocidades horizontal e vertical foram consideradas como tendo um componente aleatório. As figuras a seguir mostram a evolução das partículas.

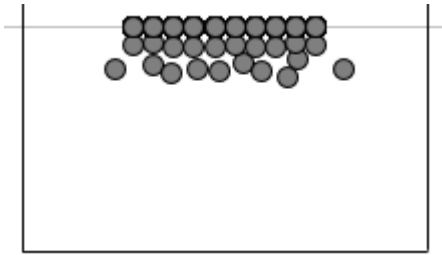


Figura 14. Quadrado $t = 0,1s$

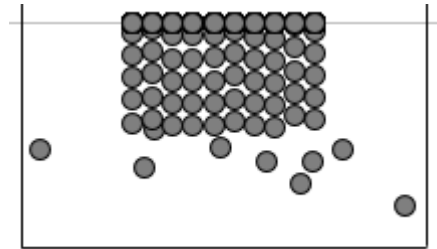


Figura 15. Quadrado $t = 0,2s$

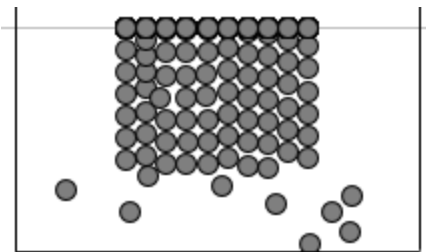


Figura 16. Quadrado $t = 0,3s$

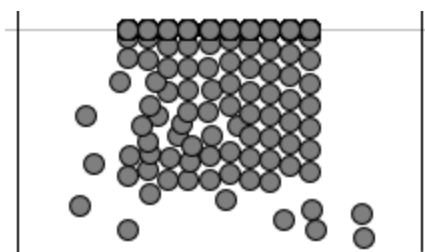


Figura 17. Quadrado $t = 0,35s$

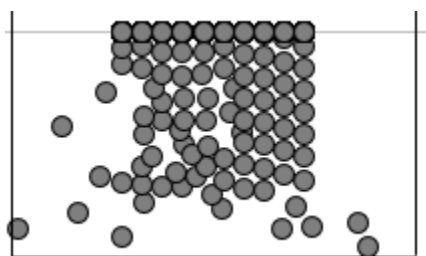


Figura 18. Quadrado $t = 0,4s$

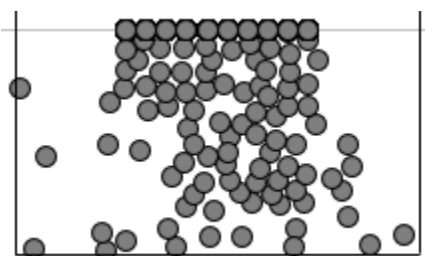


Figura 19. Quadrado $t = 0,5s$

Para este caso, como pode ser visto nas 3 primeiras imagens, o aspecto aleatório das condições iniciais, fez com que apenas algumas partículas se distanciassem das demais. Isso foi suficiente para que, nos instantes seguintes, a colisão destas com as paredes do recipiente e a posterior colisão com as outras partículas, deixa-se o comportamento do sistema como um todo, caótico.

ii. ANTEPAROS INCLINADOS

No caso dos anteparos inclinados, o único componente aleatório que foi simulado, foi a velocidade vertical, mantendo-se a velocidade inicial horizontal nula.

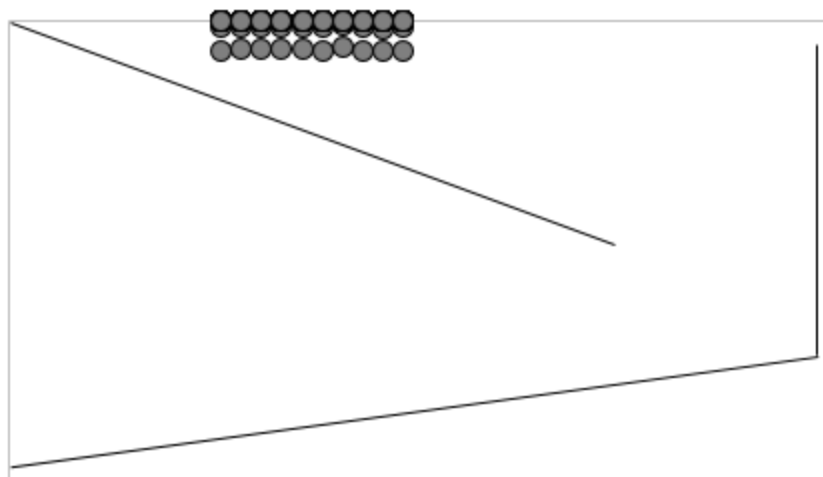


Figura 20. Anteparos $t = 0,1s$

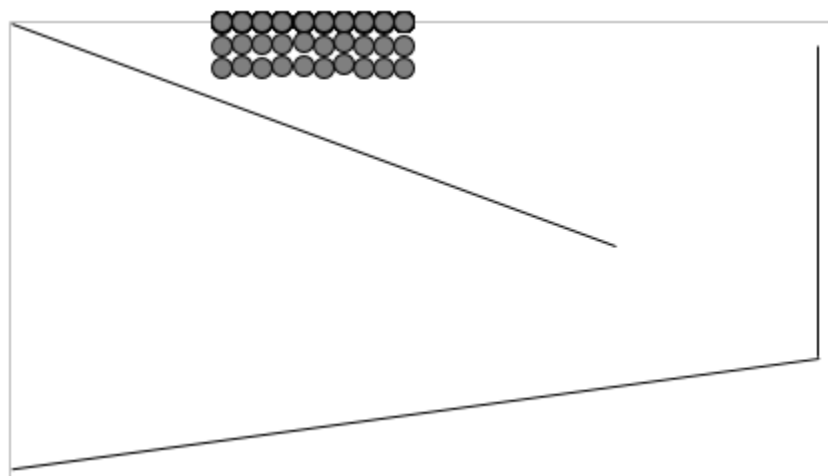


Figura 21. Anteparos $t = 0,2s$

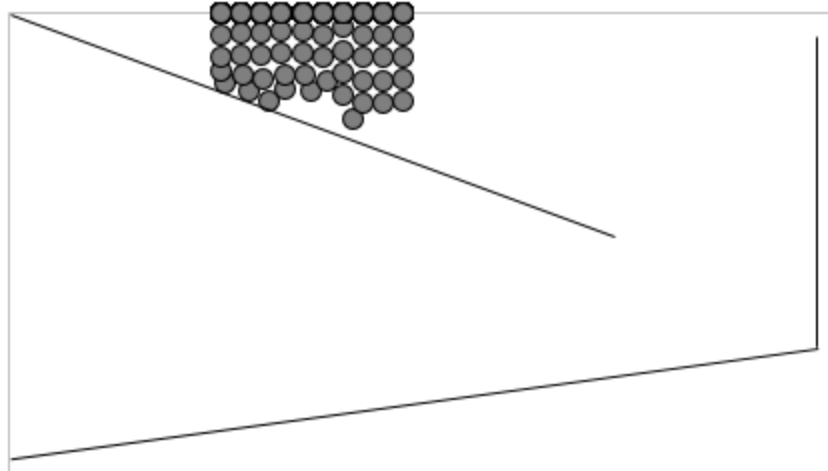


Figura 22. Anteparos $t = 0,3s$

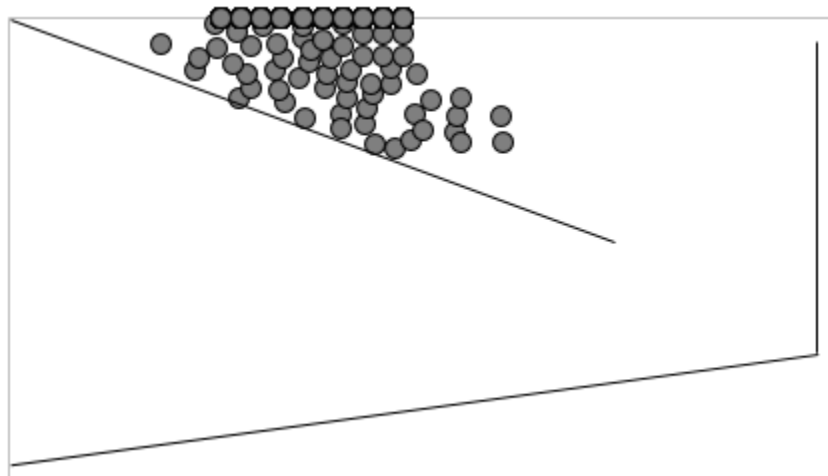


Figura 23. Anteparos $t = 0,4s$

Para este caso, como pode ser visto nas duas primeiras imagens da simulação, o sistema se comporta de forma ordenada, imediatamente antes da colisão com o anteparo. Nos instantes posteriores, a colisão das partículas com o anteparo e entre elas, faz com que o sistema se torne caótico e as posições finais de difícil previsão.

CAPÍTULO VI

CONCLUSÕES E PERSPECTIVAS

No presente trabalho foi apresentado uma forma de aplicação do Método dos Elementos Discretos, por um algoritmo escrito em Excel VBA, que mostrou a influência das condições de contorno no arranjo das partículas.

As dificuldades encontradas no presente trabalho foram as de adequar os parâmetros numéricos para representar de forma razoável a realidade, como por exemplo o passo de tempo, da ordem de 10^{-6} s, juntamente com as condições iniciais e as condições de contorno.

As limitações tanto das propriedades das partículas, quanto da interação entre elas, formuladas neste trabalho, restringem a aplicação para áreas mais simples e que, portanto, não envolvam geometrias mais complexas e propriedades não uniformes, como é o caso da mineração, por exemplo.

O presente trabalho sugere como principais perspectivas:

- Mudança do método de iteração para o direct mapping;
- Inserção da força tangencial entre as partículas;
- Interação das partículas com algum tipo de fluido.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] NEVES, C. E. V. (2009) Comportamento de Materiais Granulares usando o Método dos Elementos Discretos. Dissertação de Mestrado, Publicação, G.DM-173/09, Departamento de Engenharia Civil, Universidade de Brasília, Brasília, DF, 166p.
- [2] ROBERTO DOMINGO RIOS, (2002) Aplicações do Método dos Elementos Discretos em Estruturas de Concreto.
- [3] LUIS EDUARDO KOSTESKI, Aplicação do Método dos Elementos Discretos Formado por Barras no estudo do colapso de Estruturas.
- [4] SAMUEL THIMOUNIER FERREIRA (2009) Estudo e Implementação do Método dos Elementos Discretos em 3D.
- [5] SAMPAIO, MARCO ANTÔNIO BRASIEL O método dos elementos discretos com superelipsóides usando a parametrização das rotações de Rodrigues.
- [6] DEISE M. RIBEIRO, PAULO C. CORRÊA, BRUNO F. FURTADO ET AL., Propriedades mecânicas dos grãos de soja em função do teor de água, Eng. Agríc., Jaboticabal, v.27, n.2, p.493-500, maio/ago. 2007

APÊNDICE A

A.1 ALGORITMO EM EXCEL VBA

```
Sub DEM(control As IRibbonControl)

Dim n As Long
Dim lw As Double
Dim Sh As Shape
Dim qnt_part As Long
Dim qnt As Long
Dim cont As Integer
Dim ymax As Double
Dim rng As Range

Set ws2D = ThisWorkbook.Worksheets("Modelo2D")
Set wsD = ThisWorkbook.Worksheets("Developer")
Set wsV = ThisWorkbook.Worksheets("Visualization")

Application.ScreenUpdating = False

ws2D.Range("A2:K100000").ClearContents

' Tipo de simulação:
' cond = 1 ----- Anteparos
' cond = 2 ----- Cubo
' cond = 3 ----- Funil

cond = 1

n = 1000
total_time = 5           '[s]

d_min = 0.1              '[m]
d_max = 0.1              '[m]
rho = 1.1 * 10 ^ 3       '[Kg/m^3]
u = 0.03                 '[-]

g = 9.80665              '[m/s^2]
```

```

Kn = 1.4 * 10 ^ 6           '[N/m]

d = (d_min + d_max) / 2
m = (3.1415 / 6 * d ^ 3) * rho

t_crit = (m / Kn) ^ 0.5
dt = 0.005 * t_crit

sl = 1                       'Source left
SrcWidth = 1                 'Source Width

' ----- Anteparo 1 -----
a1_p1x = 0
a1_p1y = 0
a1_p2x = 3
a1_p2y = 2
a1_alpha = Atn((a1_p2y - a1_p1y) / (a1_p2x - a1_p1x))

' ----- Anteparo 2 -----
a2_px = 4

' ----- Anteparo 3 -----
a3_p1x = 4
a3_p1y = 3
a3_p2x = 0
a3_p2y = 4
a3_alpha = Atn((a3_p1y - a3_p2y) / (a3_p1x - a3_p2x))
' -----

cont = 1
' Set conditions
With ws2D

    For i = 1 To n
        .Range("n_part").Cells(i, 1) = i
        .Range("t").Cells(i, 1) = 0
        .Range("diam").Cells(i, 1) = 0.1
    
```

```

        .Range("sj").Cells(i, 1) = 0
        .Range("vi").Cells(i, 1) = 3
        .Range("vj").Cells(i, 1) = Round(3 + Rnd() * 0.5, 6)
        .Range("ai").Cells(i, 1) = 0
        .Range("aj").Cells(i, 1) = g
    Next i

    ' quantidade de particulas em cada fileira (arredondado pra baixo)
    qnt_part = Application.WorksheetFunction.RoundDown(SrcWidth / d, 0)
    qnt = qnt_part
    ' quantidade de vezes até chegar a n
    n_times = Application.WorksheetFunction.RoundDown(n / Application.WorksheetFunction.RoundDown(SrcWidth / d, 0), 0)

    ' Set si0
    For j = 1 To n_times
        .Range("si").Cells(1 + (j - 1) * qnt_part, 1) = s1 + .Range("diam").Cells(1 + (j - 1) * qnt_part, 1) / 2
        For k = 2 To qnt_part
            .Range("si").Cells(k + (j - 1) * qnt_part, 1) = .Range("si").Cells(k + (j - 1) * qnt_part - 1, 1) + _
                .Range("diam").Cells(k + (j - 1) * qnt_part - 1, 1) / 2 + _
                .Range("diam").Cells(k + (j - 1) * qnt_part, 1) / 2
        Next k
        If j = n_times Then
            b = 1 + n_times * qnt_part
            .Range("si").Cells(1 + n_times * qnt_part, 1) = s1 + .Range("diam").Cells(1 + n_times * qnt_part, 1) / 2
            For l = 2 To n - qnt_part * n_times
                .Range("si").Cells(l + n_times * qnt_part, 1) = .Range("si").Cells(l + n_times * qnt_part - 1, 1) + _
                    .Range("diam").Cells(l + n_times * qnt_part - 1, 1) / 2 + _
                    .Range("diam").Cells(l + n_times * qnt_part, 1) / 2
            Next l
        End If
    Next j

```

```

End If
Next j

End With

For t = dt To total_time Step dt

    For i = 1 To qnt_part

        ' ----- linha da partícula -----
        linha_part = Application.WorksheetFunction.Match(i, ws2D.Columns("A"), 0) - 1
        If ws2D.Range("t").Cells(linha_part, 1) = 0 And t > dt Then GoTo next_part:

        ' ----- Condições do passo de tempo anterior -----
        si0 = ws2D.Range("si").Cells(linha_part, 1)
        sj0 = ws2D.Range("sj").Cells(linha_part, 1)
        vi0 = ws2D.Range("vi").Cells(linha_part, 1)
        vj0 = ws2D.Range("vj").Cells(linha_part, 1)
        ai0 = ws2D.Range("ai").Cells(linha_part, 1)
        aj0 = ws2D.Range("aj").Cells(linha_part, 1)
        Fni = 0
        Fnj = 0

        ws2D.Range("J2").Formula = "=ROUND(SQRT((D2-$D$" & linha_part + 1 & ")^2+(E2-$E$" & linha_part + 1 & ")^2), 3)"
        ws2D.Range("K2").Formula = "=IF(J2=0,FALSE,J2<(C2+$C$" & linha_part + 1 & ")/2)"
        ws2D.Range("J2:K2").AutoFill Destination:=ws2D.Range("J2:K" & qnt_part + 1)

        ' ----- partícula-partícula -----
        cont_contato = Application.WorksheetFunction.CountIf(ws2D.Columns("K"), True)
        If cont_contato > 0 Then
            linha = 1
            For c = 1 To cont_contato

                linha = ws2D.Columns("K").Find(What:=True, LookIn:=xlValues, _
                    after:=ws2D.Range("K" & linha)).Row
            
```

```

        Fn = Kn * ws2D.Range("J" & linha) / 2

        x      =      ws2D.Range("si").Cells(linha_part,      1)      -
ws2D.Range("si").Cells(linha - 1, 1)
        y      =      ws2D.Range("sj").Cells(linha_part,      1)      -
ws2D.Range("sj").Cells(linha - 1, 1)
        dist_centro = ws2D.Range("J" & linha)

        Fni = Fn * x / dist_centro
        Fnj = Fn * y / dist_centro

    Next c
End If

Select Case cond

    Case 1:

        ' ----- partícula-parede -----
        ' Anteparo 1
        al_alpha1 = Atn((sj0 - a1_ply) / si0)
        dist = (((si0 - a1_plx) ^ 2 + (sj0 - a1_p2y) ^ 2) ^ 0.5) * Sin(al_alpha
- al_alpha1)
        If dist < (ws2D.Range("diam").Cells(linha_part, 1) / 2) And sj0 >= a1_ply
Then

            Fn = Kn * dist
            Fat = u * Fn

            Fni = Fni + Fn * Sin(al_alpha)
            Fnj = Fnj - Fn * Cos(al_alpha)

        End If

        ' Anteparo 2
        dist = a2_px - (si0 + ws2D.Range("diam").Cells(linha_part, 1) / 2)
        If dist < 0 Then

```

```

        Fni = Kn * dist
End If

' Anteparo 3
a3_alpha3 = Atn((sj0 - a1_ply) / (a3_p1x - si0))
dist = (((si0 - a1_p1x) ^ 2 + (sj0 - a1_p2y) ^ 2) ^ 0.5) * Sin(a3_alpha
- a3_alpha3)

If dist < (ws2D.Range("diam").Cells(linha_part, 1) / 2) And sj0 >= a3_ply
Then

        Fn = Kn * dist
        Fat = u * Fn

        Fni = Fni + Fn * Sin(a1_alpha)
        Fnj = Fnj - Fn * Cos(a1_alpha)

End If
' -----

Case 2:

' Anteparo Horizontal
dist = 2 - (sj0 + ws2D.Range("diam").Cells(linha_part, 1) / 2)
If dist < 0 Then
        Fnj = Kn * dist
End If

' Anteparo Vertical Esquerda
dist = si0 - 0.5 - (ws2D.Range("diam").Cells(linha_part, 1) / 2)
If dist < 0 Then
        Fni = -Kn * dist
End If

' Anteparo Vertical Direita
dist = 2.5 - si0 - (ws2D.Range("diam").Cells(linha_part, 1) / 2)
If dist < 0 Then
        Fni = Kn * dist
End If

```

Case 3:

```
' Anteparo 1
a1_alpha1 = Atn((sj0 - a1_p1y) / si0)
dist = (((si0 - a1_p1x) ^ 2 + (sj0 - a1_p2y) ^ 2) ^ 0.5) * Sin(a1_alpha
- a1_alpha1)

If dist < (ws2D.Range("diam").Cells(linha_part, 1) / 2) And sj0 >= a1_p1y
Then

    Fn = Kn * dist
    Fat = u * Fn

    Fni = Fni + Fn * Sin(a1_alpha)
    Fnj = Fnj - Fn * Cos(a1_alpha)

End If

End Select

' Atualiza os valores
ws2D.Range("t").Cells(linha_part, 1) = t

' Direção x
ai = Fni / m
vi = vi0 + ai * dt
si = si0 + vi0 * dt + ai * dt ^ 2 / 2

' Direção y
aj = Fnj / m + g
vj = vj0 + aj * dt
sj = sj0 + vj0 * dt + aj * dt ^ 2 / 2

ws2D.Range("ai").Cells(linha_part, 1) = ai
ws2D.Range("vi").Cells(linha_part, 1) = vi
ws2D.Range("si").Cells(linha_part, 1) = si
ws2D.Range("aj").Cells(linha_part, 1) = aj
ws2D.Range("vj").Cells(linha_part, 1) = vj
```

```

ws2D.Range("sj").Cells(linha_part, 1) = sj

c = ws2D.Range("sj").Cells(linha_part, 1)
d = (ws2D.Range("diam").Cells(linha_part, 1) +
ws2D.Range("diam").Cells(linha_part + qnt, 1))

' Verifica se a posição vertical das partículas já é suficiente para que novas
partículas possam ser inseridas na iteração
If ws2D.Range("sj").Cells(linha_part, 1) >= ws2D.Range("diam").Cells(linha_part,
1) + ws2D.Range("diam").Cells(linha_part + qnt, 1) And ws2D.Range("t").Cells(linha_part
+ qnt, 1) = 0 Then
    ws2D.Range("t").Cells(linha_part + qnt, 1) = t
    ws2D.Range("vi").Cells(linha_part + qnt, 1) = 3
    ws2D.Range("vj").Cells(linha_part + qnt, 1) =
ws2D.Range("vj").Cells(linha_part, 1) * (1 + 0.01 * (0.5 - Rnd()))
    If linha_part + qnt > qnt_part Then qnt_part = linha_part + qnt
End If

next_part:

Next i

Next t

End Sub

Sub teste2()
For i = 1 To n
    For j = i + 1 To n
        'Calculo da colisão entre os elementos i e j
    Next j
Next i
End Sub

```